ByteDance | Seed

ICLR

# Coupling Experts and Routers in Mixture-of-Experts via an Auxiliary Loss

**Ang Lv**, Jin Ma, Yiyuan Ma, Siyuan Qiao

April, 2026

ByteDance | Seed

ICLR

# Background

**Mixture-of-Experts (MoE)** uses standalone routers to select experts for each token.

MoEs suffer from *the separation* between the **router's decision-making** and the **experts' execution**.

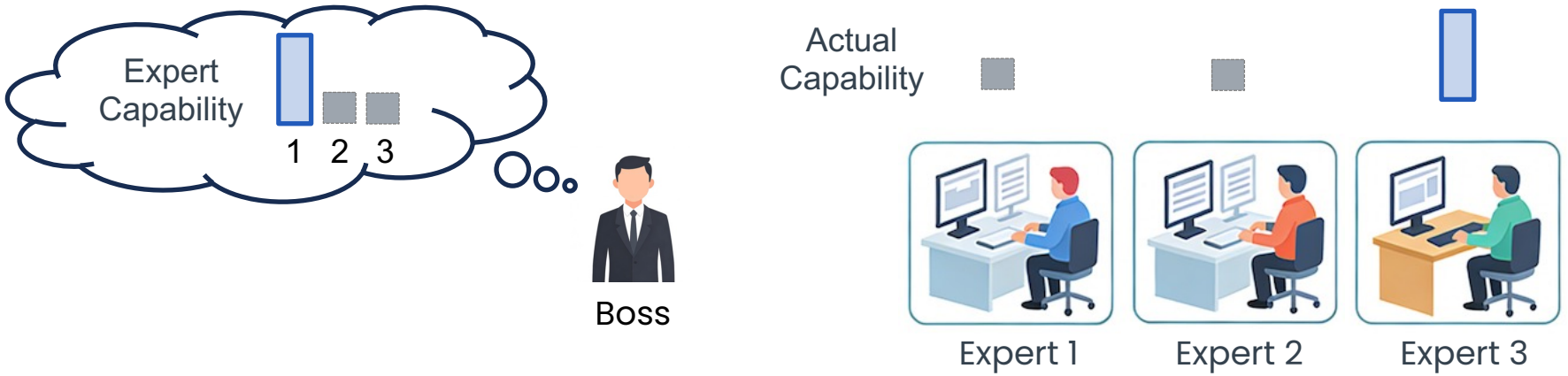



Figure: The separation between decision-making and execution.

ByteDance | Seed

ICLR

# Background

**Mixture-of-Experts (MoE)** uses standalone routers to select experts for each token.

MoEs suffer from *the separation* between the **router's decision-making** and the **experts' execution**.

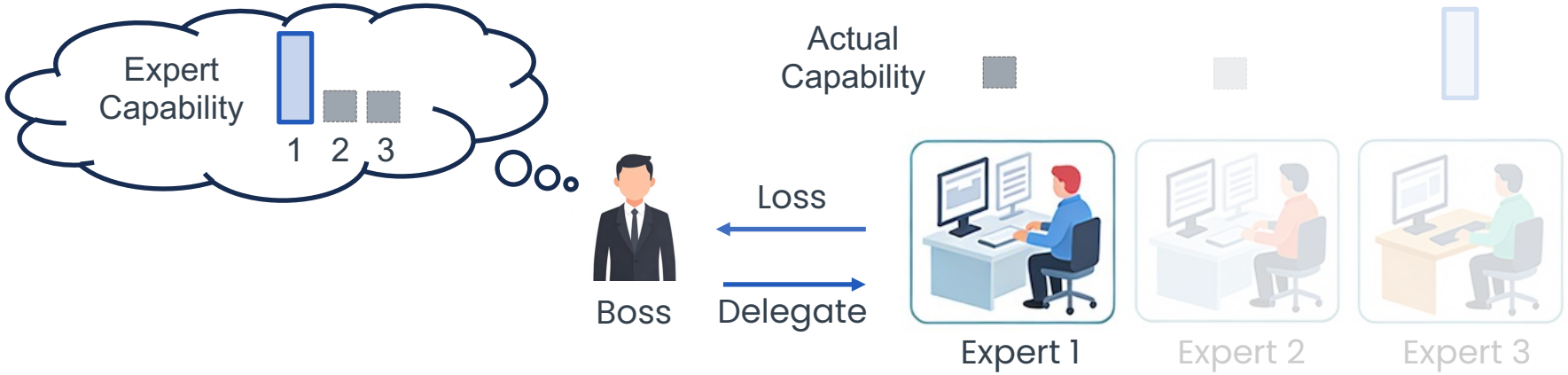



Figure: The separation between decision-making and execution.

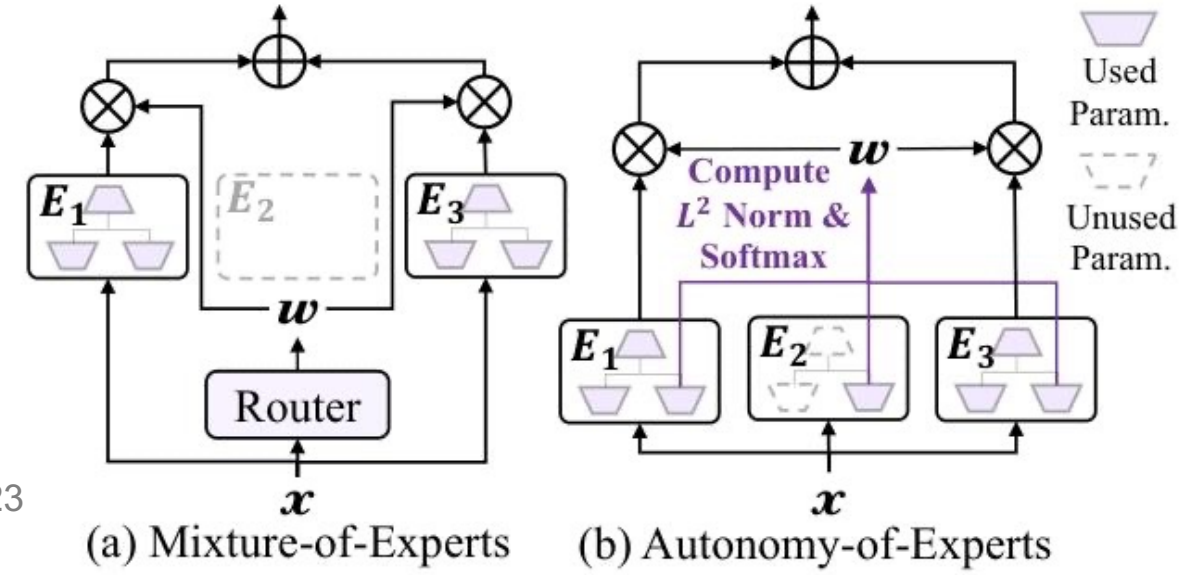
# Previous Works

**An insight:** A module matching the input's requirements yields high activations. (Section 4.1, [1])

Inspired by this, **Autonomy-of-Experts (AoE)** [2] addresses the "separation issue":

Each token is fed to all experts, and only the top-K experts with the highest intermediate activation norms are selected.

AoE is impractical due to its exhaustive computational complexity, which scales linearly with the number of tokens.




[1] Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time, ICML 2023
[2] Autonomy-of-Experts Models, ICML 2025

# Design Principles

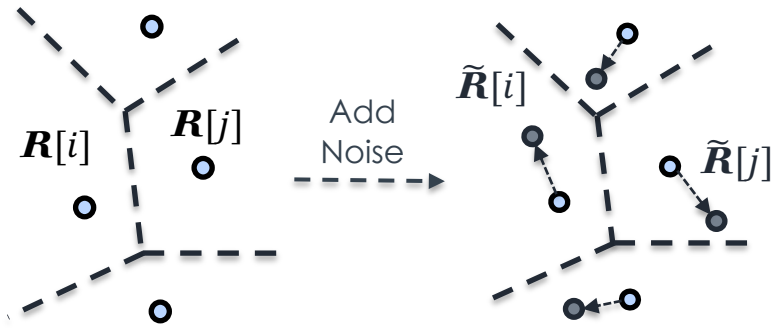
1. Routers must be retained in MoE architectures to preserve routing efficiency.

2. An auxiliary loss that enables interaction between experts and routers can strengthen their coupling.

3. The loss must have complexity independent of the number of input tokens.

# Expert-Router Coupling Loss

There are three steps to calculate ERC loss, which efficiently addresses the separation issue.




Router parameters $\boldsymbol{R} \in \mathbb{R}^{n \times d}$ are cluster centers.

Step 1: Perturb $\boldsymbol{R}$ as proxy tokens:

$$\tilde{\boldsymbol{R}}[i] = \boldsymbol{R}[i] \odot \boldsymbol{\delta}_i. \quad \boldsymbol{\delta}_i \in \mathbb{R}^d$$

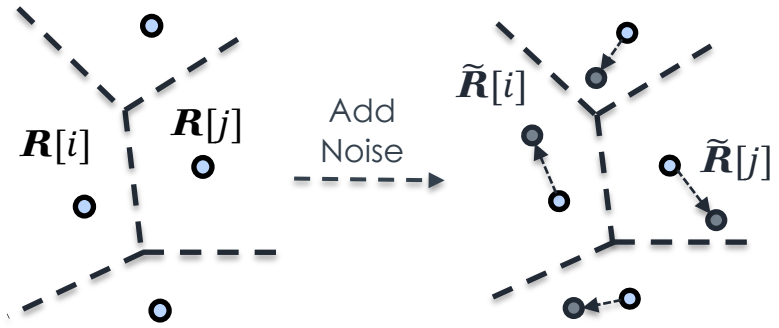
$$\boldsymbol{\delta}_i \sim \mathcal{U}(1-\epsilon_i, 1+\epsilon_i)^d.$$

$$\epsilon_i \leq \frac{\|\boldsymbol{R}[i] - \boldsymbol{R}[j]\|}{2\|\boldsymbol{R}[i]\|}. \qquad j = \arg\min_{j^* \neq i} \|\boldsymbol{R}[i] - \boldsymbol{R}[j^*]\|$$

$\tilde{R}[i]$ is a proxy for tokens routed to expert $i$.

# Expert-Router Coupling Loss

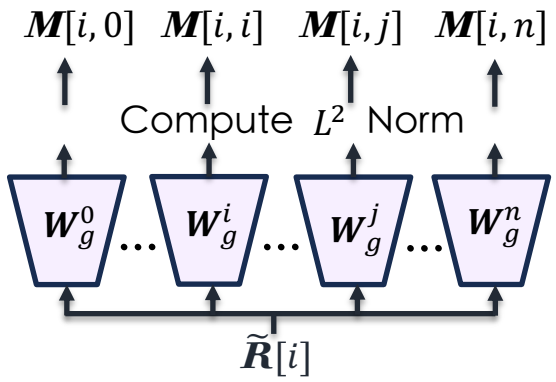
There are three steps to calculate ERC loss, which efficiently addresses the separation issue.




Router parameters are cluster centers.

Step 1: Perturb $\boldsymbol{R}$ as proxy tokens.

$\widetilde{\boldsymbol{R}}[i]$ is a proxy for tokens routed to expert $i$.

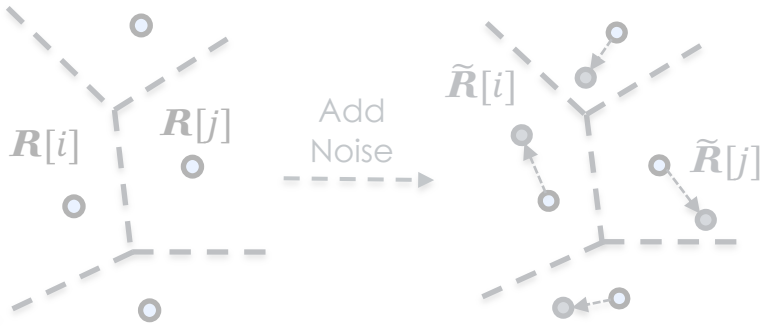



Step 2: Input all $\widetilde{\boldsymbol{R}}[i]$ to all experts; $\boldsymbol{W}_g^i \in \mathbb{R}^{d \times D}$

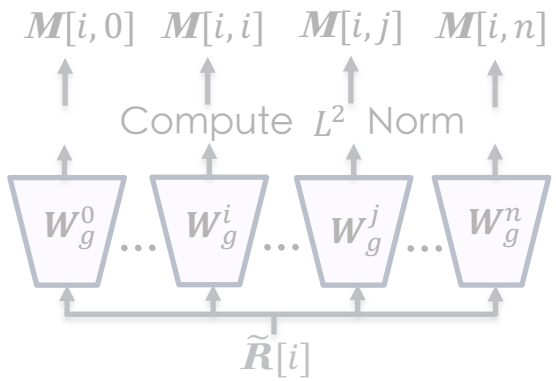
Compute $\boldsymbol{M} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{n}}$, the $L^2$ norm of the intermediate activations.

$\boldsymbol{M}[i,j]$ is the activation norm produced by expert $j$ given $\widetilde{\boldsymbol{R}}[i]$.

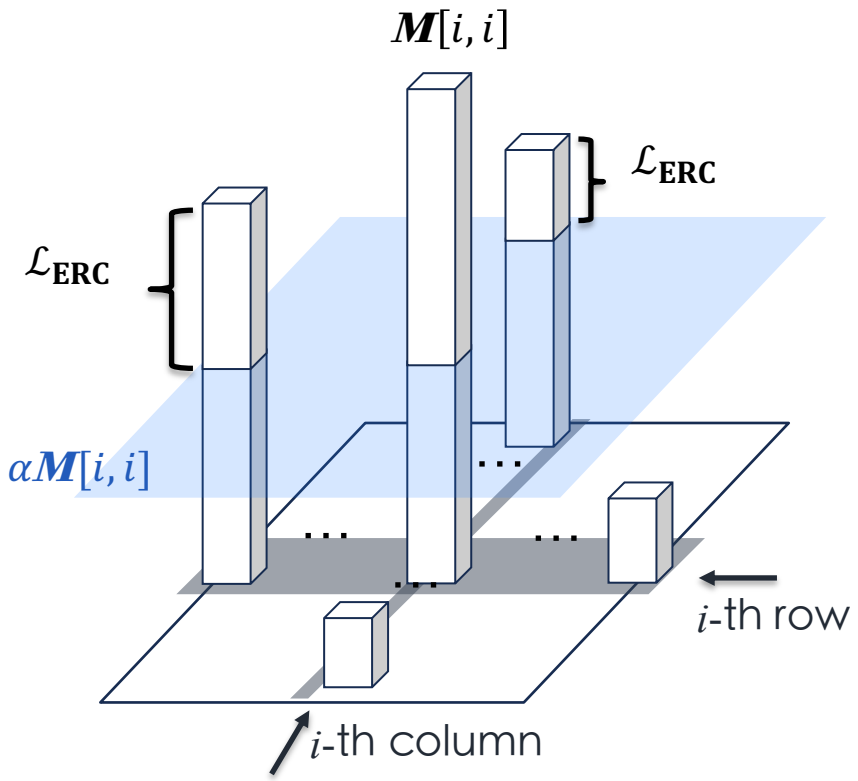
# Expert-Router Coupling Loss




Step 1: Perturb $\boldsymbol{R}$ as proxy tokens.




Step 2: Input all $\widetilde{\boldsymbol{R}}[i]$ to all experts;
Compute $\boldsymbol{M} \in \mathbb{R}^{n \times n}$.

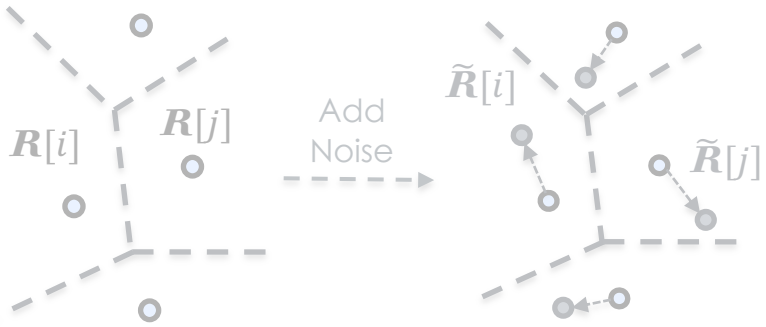



Step3: The ERC loss enforces

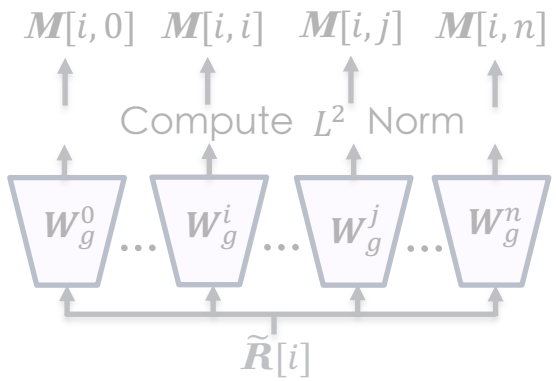
$$(1)\ \boldsymbol{M}[i,j] < \alpha\, \boldsymbol{M}[i,i]$$
$$(2)\ \boldsymbol{M}[j,i] < \alpha\, \boldsymbol{M}[i,i]. \qquad \forall i, j \neq i$$

$\boldsymbol{M}[i,j]$ is the activation norm produced by expert $j$ given $\widetilde{\boldsymbol{R}}[i]$.

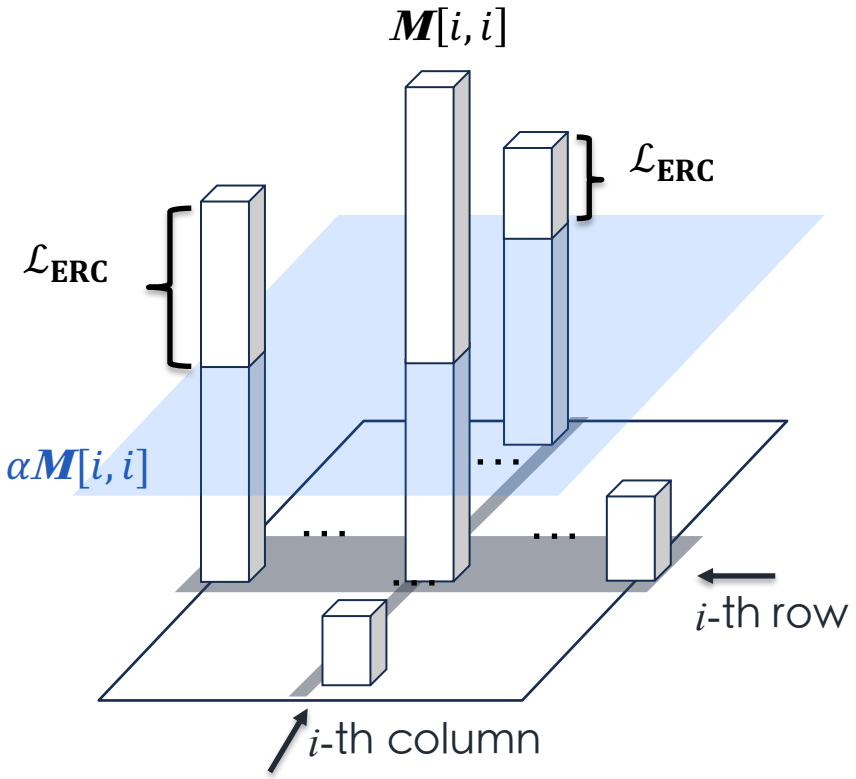
# Expert-Router Coupling Loss




Step 1: Perturb $\boldsymbol{R}$ as proxy tokens.




Step 2: Input all $\tilde{\boldsymbol{R}}[i]$ to all experts;
Compute $\boldsymbol{M} \in \mathbb{R}^{n \times n}$.




Step3: The ERC loss enforces

$$(1)\ \boldsymbol{M}[i,j] < \alpha\, \boldsymbol{M}[i,i] \quad \forall i, j \neq i$$
$$(2)\ \boldsymbol{M}[j,i] < \alpha\, \boldsymbol{M}[i,i].$$

$$\mathcal{L}_{\mathrm{ERC}} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left( \max\left(\boldsymbol{M}[i,j] - \alpha \boldsymbol{M}[i,i], 0\right) + \max\left(\boldsymbol{M}[j,i] - \alpha \boldsymbol{M}[i,i], 0\right) \right).$$

*$n^2$ activations only, token count independent!*

# Efficiency

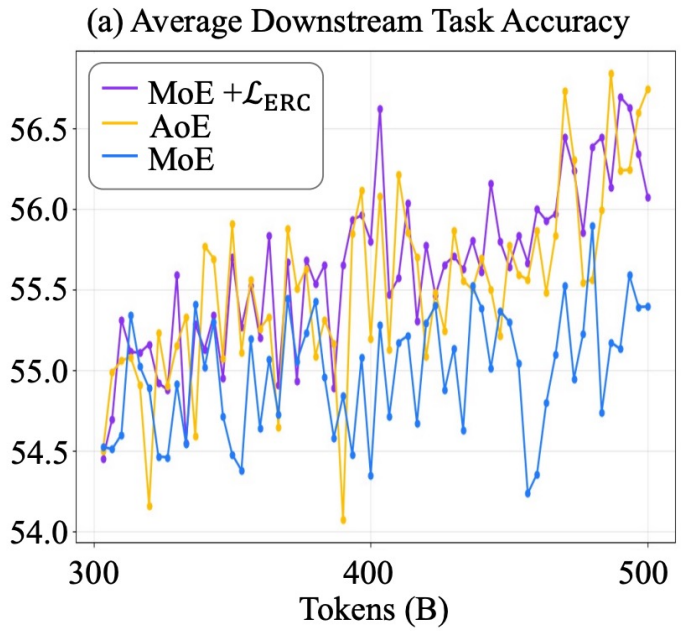
Additional FLOPs vs. Standard MoE, $\boldsymbol{T}$ (millions) tokens, $\boldsymbol{K}$ out of $\boldsymbol{n}$ (hundreds) experts, $\boldsymbol{W}_g^i \in \mathbb{R}^{d \times D}$

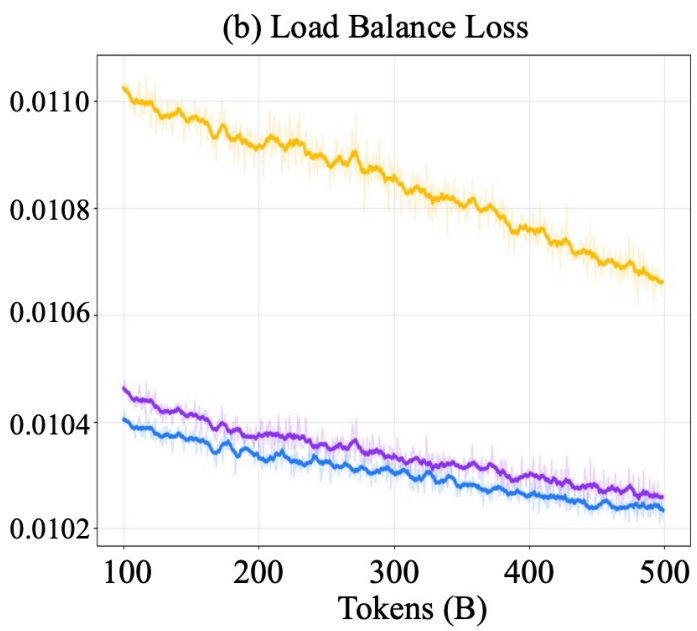
- AoE: $2\boldsymbol{T}Dd(n-K)$
- ERC loss: $2n^2Dd$, and features ***no inference overhead.*** $\boldsymbol{O}(\boldsymbol{Tn})$ $\boldsymbol{O}(\boldsymbol{n^2})$

Empirically, ERC loss results in a slight decrease in training throughput, with a magnitude of 0.2% to 0.8%.

# Experiments

3B parameters (8 out of 64 experts), trained on 500B tokens, task accuracy averaged across 10 tasks.




15B parameters (8 out of 256 experts), trained on 500B tokens, task accuracy averaged across 10 tasks.

| | MMLU | C-Eval | MMLU-Pro | AGI-Eval | BBH | MATH | GSM8K | TriviaQA |
|---|---|---|---|---|---|---|---|---|
| MoE | 63.2 | 67.5 | 31.0 | 42.0 | 44.3 | 25.7 | 45.2 | 47.2 |
| MoE + $\mathcal{L}_{\text{ERC}}$ | 64.6 | 69.0 | 31.9 | 44.2 | 45.6 | 26.1 | 45.8 | 49.1 |

# ERC loss and Specialization

With the ERC loss, experts are more specialized, as they exhibit greater discrimination between tokens they process and those they do not.

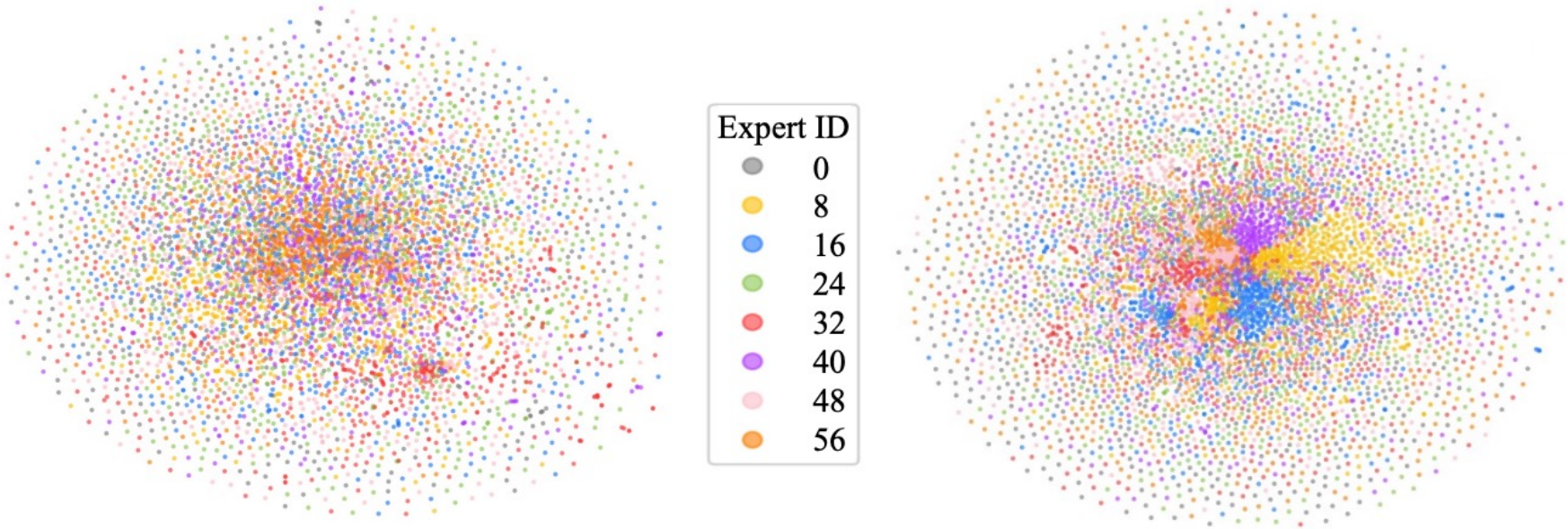



**Figure 4** t-SNE projections of $\boldsymbol{W}_g$ in MoE experts trained without and with the ERC loss. Our ERC loss provides greater expert specialization.

ByteDance | Seed

ICLR

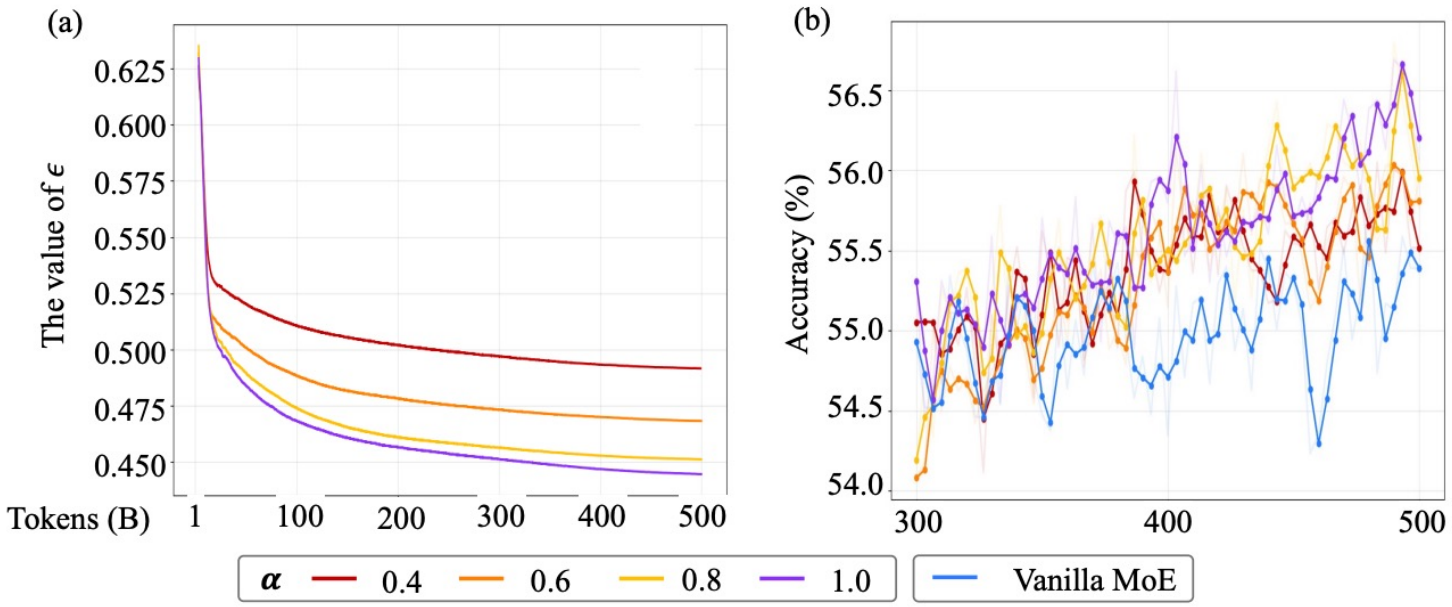
# ERC loss and Specialization

The ERC loss is an effective tool for exploring expert specialization.

(a) Since routers are deeply coupled with experts, the distance between neighboring cluster centers (i.e., the noise level $\epsilon$) quantitatively reflects changes in expert specialization during training, which is controlled by $\alpha$.

Recap: $\epsilon_i \leq \frac{\|\boldsymbol{R}[i] - \boldsymbol{R}[j]\|}{2\|\boldsymbol{R}[i]\|}. \qquad j = \arg\min_{j^* \neq i} \|\boldsymbol{R}[i] - \boldsymbol{R}[j^*]\|$

(b) Downstream performance across different values of $\alpha$.



# Unresolved Challenges

1. Currently, our only way to explore the optimal specialization degree is through hyperparameter tuning (e.g., α), with performance on downstream tasks as the feedback signal.

2. Numerous factors complicate the identification of the optimal degree:

- K and n
- Shared expert design
- ……

3. Is expert output orthogonality a desirable goal?

ByteDance | Seed

ICLR

# Thank you

# Q & A

Welcome to contact me: anglv@ruc.edu.cn

Visit my homepage: trestad.github.io